307-CD-003-003
329-CD-003-003

EOSDIS Core System Project

Communications and Systems Management Segment (CSMS) Release and Development Plan for the ECS Project

Final

June 1995

Hughes Applied Information Systems
Landover, Maryland

Communications Systems and Systems Management Segment (CSMS) Release and Development Plan for the ECS Project

June 1995

Prepared Under Contract NAS5-60000
CDRL Items 048, 058

SUBMITTED BY

E. M. Lerner 6/6/95

Edward Lerner, Segment Manager Date
EOSDIS Core System Project

Hughes Applied Information Systems
Landover, Maryland

307-CD-003-003
329-CD-003-003

This page intentionally left blank.

Preface

This document is a formal contract deliverable with an approval code 2. As such, it does not require formal Government approval, however, the Government reserves the right to request changes within 45 days of the initial submittal or any subsequent revision. Changes to this document shall be made by document change notice (DCN) or by complete revision.

Once approved, this document shall be under ECS Project Configuration Control.

Any questions should be addressed to:

Data Management Office
The ECS Project Office
Hughes Applied Information Systems
1616 McCormick Dr.
Landover, MD 20785

This page intentionally left blank.

Abstract

The CSMS Release and Development Plan satisfies the requirements for CDRL Items 048, DID 307/DV2 (Segment Release Plan) and 058, DID 329/DV2 (Segment Development Plan), as specified in the Statement of Work, as a deliverable under the Earth Observing System Data and Information System (EOSDIS) Core System (ECS) Contract NAS5-60000.

This document describes the plan for development of the Configuration Items (CIs) and components of the Communications and System Management Subsystem (CSMS) of the ECS. The ECS is deployed as a series of releases, each providing additional functionality, in support of scheduled key EOSDIS element deployment, and performance enhancements as planned technologies mature. Each release contains a subset of the functionality specified in ECS Functional and Performance Requirements Specification (F&PRS), with the final ECS release containing all of functionality specified for the program. This version of the CSMS Development/Release Plan includes details of the CSMS development for Interim Release 1 (IR-1), and Release A of the ECS. Subsequent versions are planned for release at the IDRs for Releases B through D.

Keywords: CSMS, Development, Release, Schedule, Configuration Item, Component, Software Lines of Code, Detailed Design, Code and Unit Test, Integration and Test

This page intentionally left blank.

Change Information Page

List of Effective Pages	
Page Number	**Issue**
Title	Final
iii through xii	Final
1-1 and 1-2	Final
2-1 and 2-2	Final
3-1 through 3-4	Final
4-1 through 4-10	Final
5-1 through 5-8	Final
6-1 through 6-6	Final
7-1 and 7-2	Final
8-1 and 8-2	Final
9-1 and 9-2	Final
10-1 and 10-2	Final
A-1 through A-4	Final
AB-1 through AB-3	Final

Document History			
Document Number	**Status/Issue**	**Publication Date**	**CCR Number**
307-CD-003-001 329-CD-003-001	Review Copy	February 1995	
307-CD-003-002 329-CD-003-002	Final	March 1995	95-0156
307-CD-003-003 329-CD-003-003	Final	June 1995	95-0379A
307-CD-003-003 329-CD-003-003	Final	June 1995	95-0379B

This page intentionally left blank.

Contents

Preface

Abstract

1. Introduction

1.1 Identification 1-1

1.2 Scope 1-1

1.3 Purpose 1-1

1.4 Status and Schedule 1-1

1.5 Organization 1-2

2. Related Documentation

2.1 Parent Documents 2-1

2.2 Applicable Documents 2-1

3. Development and Release Process

3.1 Component Identification 3-1

3.2 Schedule Preparation 3-2

3.3 Integration and Test 3-3

3.4 Product Delivery 3-3

4. Developed Component Definitions

4.1 CSMS Components 4-1

4.2 CSMS Development Estimation 4-7

4.2.1 Custom Software Estimation Modeling 4-8

4.2.2 Hybrid Software Development 4-9

4.2.3 COTS-based Development 4-9

5. IR-1 Development Schedules

5.1 IR-1 Objectives 5-1

5.2 IR-1 Development 5-1

5.2.1 IR-1 Detailed Design 5-3

5.2.2 COTS Requirements 5-3

5.2.3 COTS Selection 5-3

5.2.4 COTS Procurement 5-4

5.2.5 COTS Installation and Checkout 5-5

5.2.6 EP & Prototype Reuse 5-5

5.2.7 IR-1 Custom Software Code and Unit Test 5-5

5.2.8 IR-1 COTS Software Configuration and Hybrid Software Development 5-5

5.2.9 CSMS IR-1 I & T Development and Test 5-6

5.3 IR-1 Components Mapping to Threads and Builds 5-6

6. Release A Development Schedules

6.1 Release A Objectives 6-1

6.2 Release A Development 6-1

6.2.1 Detailed Design 6-3

6.2.2 COTS Requirements 6-3

6.2.3 COTS Selection 6-3

6.2.4 COTS Procurement 6-4

6.2.5 Release A COTS Configuration/Hybrid Software Development 6-4

6.2.6 Release A Custom Software Code and Unit Test 6-4

6.2.7 Release A Integration and Test 6-4

6.3 Release A Components Mapping to Threads and Builds 6-4

7. Release B Development Schedules

8. Release C Development Schedules

9. Release D Development Schedules

10. Product Delivery

10.1 General 10-1
10.2 CSMS Products at IR-1 and Release A 10-1

Figures

5-1. Interim Release 1 Development Plan 5-2
5-2. Component Mapping to the Build/Thread Plan 5-8
6-1. CSMS Release A Development Plan 6-2

Tables

4-1. CSMS Components, IR-1 4-2
4-2. CSMS S/W Components, Release A 4-4
4-3. CSMS Deployed Hardware Components, IR-1 4-6
4-4. CSMS Deployed Hardware Components, Release A 4-7
4-5. CSMS Development Activities 4-8
5-1. CSMS Development Estimates for Interim Release 1 5-7
6-1. Development Estimates for Release A 6-5

Appendix A: Basis of Estimate for Development Effort

Abbreviations and Acronyms

This page intentionally left blank.

1. Introduction

1.1 Identification

The Communications and System Management Segment (CSMS) Release and Development Plan for the ECS Project, Contract Data Requirement List (CDRL) Items 048 and 058, with requirements specified in the Data Item Descriptions (DIDs) 307/DV2 and 329/DV2 is a required deliverable under the Earth Observing System Data and Information System (EOSDIS) Core System (ECS), Contract NAS5-60000.

1.2 Scope

This document describes the plan for development of the Configuration Items (CIs) and components of the Communications and Systems Management Segment (CSMS) of the Earth Observing System (EOS) Data Information System (EOSDIS) Core System (ECS). The ECS is deployed as a series of releases, each providing additional functionality, in support of scheduled key EOSDIS element deployment, and performance enhancements, as planned technologies mature. Each release contains a subset of the functionality specified in ECS Functional and Performance Requirements Specification (F&PRS), with the final ECS release containing all of functionality specified for the program. This version of the CSMS Development/Release Plan includes details of the CSMS development for Interim Release 1 (IR-1), and Release A of the ECS. Subsequent versions are planned for release at the IDRs for Releases B through D.

This document reflects the Technical Baseline submitted via contract correspondence no. ECS 194-00343.

1.3 Purpose

This plan orchestrates the procedures defined in the ECS Software Development Plan, CDRL # 049, DID 308, into release-specific, development plans of schedule providing guidance in the preparation of the detailed planning necessary to ensure a graceful transformation from the design and prototyping activities into tangible end items ready for system integration and test. It identifies the CIs and their components; defines the resources required for component development; provides schedule templates for development, by release, and provides the mapping of components to builds by release at the ECS segment level. Specific details of the component development, coding standards, integration and test, and related items can be found in the supporting documentation listed in Section 2.2, Applicable Documents.

1.4 Status and Schedule

This version of the document addresses the general development approach for the CSMS and provides specific details for the development of the CSMS builds for Interim Release 1 and Release A of the ECS. Future updates are planned for release at the IDRs for the remaining ECS releases.

1.5 Organization

This document is organized into twelve sections:

Section 1 Introduction, introduced EOSDIS and this document.

Section 2 Related Documentation, contains a list of documents which influence or embellish the material contained in the CSMS Release and Development Plan.

Section 3 Development and Release Process, contains a description of the CSMS development and release process employed by this plan.

Section 4 Developed Component Definitions, contains a consolidated list of all CSMS CIs, and subordinate components and descriptions of all identified CIs and their components, how they are developed, and when they will be implemented.

Section 5 IR-1 Development Schedules, contains the schedules for development of the CIs and components for CSMS Interim Release 1.

Section 6 Release A Development Schedules, contains the schedules for development of the CIs and components for CSMS Release A.

Sections 7 through 9 Releases B through D, will contain the schedules for development of the CIs and components for CSMS Releases B through D.

Section 10 Product Delivery, delineates the Segment I&T builds by release, including the components which comprise each build.

Appendix A: Basis of Estimate for Development Effort

Abbreviations and Acronyms List

2. Related Documentation

2.1 Parent Documents

The parent document is the document from which this CSMS Release and Development Plan's scope and content are derived.

423-41-01	Goddard Space Flight Center, EOSDIS Core System (ECS) Statement of Work

2.2 Applicable Documents

The following documents are referenced within this CSMS Release and Development Plan, or are directly applicable, or contain policies or other directive matters that are binding upon the content of this volume.

304-CD-003-002	Communications and System Management Segment (CSMS) Requirements Specification for the ECS Project
305-CD-003-002	Communications and System Management Segment (CSMS) Design Specification for the ECS Project
308-CD-001-003	Software Development Plan for the ECS Project
319-CD-003-002	Communications and System Management Segment (CSMS) Integration and Test Plan for the ECS Project, Volume 1: IR-1
319-CD-004-002	Communications and System Management Segment (CSMS) Integration and Test Plan for the ECS Project, Volume 2: Release A
194-WP-904-002	Multi-Ttrack Development for the ECS Project
222-TP-003-005	Release Plan Content Description for the ECS Project
108-CD-001-XXX	ECS Intermediate Logic Network

This page intentionally left blank.

3. Development and Release Process

This section establishes the four-step process for the development of CSMS components, integration of these components into functional threads, and integration of the functional threads into release-specific builds which are tested at the segment level. The builds are then handed off to the systems test organization for integration at the ECS system-level for each release.

This four-step process includes: 1) Identification and characterization of all CSMS end-items by subsystem, CI, and component; 2) Preparation of release-specific development schedules which map the component development types, as identified in Tables 4-1 and 4-2, to development tasks and their associated integration and test threads; 3) The integration and test process, which integrates components into functional threads, which compose the segment builds and, 4) Identification of the deliverable products and their integrated components.

Step 3, the Integration & Test process, is defined for Releases IR-1 and A in the CSMS Integration and Test Plan, Volumes 1 and 2, 319-CD-002-002. Additional volumes of the Test Plan will be prepared in support of the IDRs for future ECS releases. The other steps (1,2, and 4 above) are described herein.

3.1 Component Identification

The purpose of component identification is to list all components necessary to build the CSMS, and to characterize them so that development plans and schedules may be developed. The resulting components list collates the components from the three CSMS subsystems, into release-specific build/thread-oriented development items.

Component Identification begins once the CSMS preliminary design begins to stabilize. The Configuration Items (CIs), identified at SDR, are broken down into components, and the requirements for those components are evaluated, as trade studies are performed to determine the developmental nature of each. Components have been characterized using the following type definitions in this document:

1. COTS Hardware. Includes host hardware (e.g., servers, workstations, disks, tape drives), peripherals (e.g., printers, trackballs), communications equipment (e.g., routers, hubs, switches) and test equipment.
2. COTS Software. Includes operating systems, system management and administration applications, network management and configuration tools, various utilities, software and script development tools, DBMS, libraries, management agents and management information bases (MIBs).
3. Custom Software. Includes legacy reuse, scratch-developed C/C++ software, or software developed using other ECS-approved high-level languages (HLLs).

4. Hybrid Software. Includes software that is neither custom nor COTS such as 4GL scripts; UNIX shell scripts; Motif templates; MIB tailoring data; ECS-unique configuration solutions for software packages such as HP OpenView and ClearCase; and detailed schema for DBMSs.

Software components identified during the preliminary design phase may be aggregates of 2, 3, and 4 above, and not uniquely characterized by just one type definition. Since not all COTS selections will be made until CDR, it will not been possible to characterize for example, a particular format conversion process as being COTS, custom or hybrid for all cases. Although information sufficient for scheduling purposes have been provided.

3.2 Schedule Preparation

This document defines the generic design and development processes established to produce the CSMS. These processes are used as templates for developing schedules specific to each CSMS subsystem by release. These processes will establish the underlying sequencing, dependencies, and relative time frames for the development activities to support the segment build/thread activities. They begin as the detailed designs stabilize and provide the migration path whereby prototypes and incrementally developed components are incorporated, via EP development, into the formal development and test processes. They provide the basis for detailed planning, at the work package level, used to monitor the development activities through the Performance Measurement System (PMS) to ensure a smooth transition into the integration and test phase of the program. In essence, the schedule plans herein are to be used as planning templates for official project scheduling activities.

The schedules resulting from these development templates are documented in the project Intermediate Logic Network (ILN), CDRL number 8, DID 108, which is updated monthly and maintained throughout the life of the ECS contract. The ILN consists of a series of activities and milestones, also known as nodes, which represent the major program events at the subsystem level. ILN activity nodes typically represent an aggregate of lower level work package activities which are established and monitored by the development managers. Each node is represented by name, a unique activity number, and start and finish dates. The nodes are linked together to illustrate and verify the sequencing of the activities, identify interdependencies of the nodes, and provide the basis for critical path/detailed float analysis. The ILN is typically presented by release although cross-release dependencies are also documented and maintained thus ensuring continuity of the development process through all releases.

A design and development planning template is produced for each release of CSMS to ensure that all processes, both traditional and release-specific have been identified as early as possible. The planning templates and associated descriptions for each release are placed in separate sections within this document to minimize confusion and ambiguity between releases. This version of the document contains the development templates for CSMS releases Ir1 and Release A. The plan will be updated in conjunction with the IDRs for releases B through D.

3.3 Integration and Test

As components pass the unit test phase they are submitted to the segment integration and test organization. I&T will integrate these components in the ECS Engineering Development Facility (EDF) in Landover, Maryland, where the build/thread test activities are performed. The components will be used to support functional thread development and test, leading to the integration of threads into release-specific builds. Complete details of the CSMS integration and test program, through release A, may be found in the CSMS Integration and Test Plan, Volumes 1 and 2, 319-CD-003-002.

3.4 Product Delivery

The final part of the CSMS release and development plan provides traceability of the development effort from the completed builds to components. Each release of the ECS contains increasingly more functionality, and in later releases, technology enhancements are planned. The product delivery section describes the builds and a detailed description of the components contained in them.

This page intentionally left blank.

4. Developed Component Definitions

4.1 CSMS Components

The CSMS consists of three subsystems; the Management Subsystem (MSS), the Communications Subsystem (CSS), and the Internetworking Subsystem (ISS). Each subsystem consists of one or more Configuration Items (CIs), composed of a logical grouping of software or hardware components. These components consist of Commercial-Off-the-Shelf (COTS) hardware, and custom-developed and COTS software. Many of the software components are developed by combining COTS via custom-developed software, sometimes referred to as glue code. In addition, a significant effort is made in the development of what we refer to as hybrid software, as described in Section 3.1. Collectively, the CIs provide the functionality identified in the CSMS Requirements Specification, 304-CD-003-002, although often this functionality is accomplished by two or more components of several CIs. In addition, some functionality requires the integration of components from several subsystems, including some outside of CSMS. The CSMS CIs are listed below:

CSS: Distributed Computing Software CI (DCCI)

Distributed Communications Hardware CI (DCHCI)

MSS: Management Software CI (MCI)

Management Agents CI (MACI)

Management Logistics CI (MLCI)

Management Hardware CI (MHCI)

ISS: Internetworking CI (INCI)

Internetworking Hardware CI (INHCI)

This section identifies the components used to develop the CSMS. Tables 4-1 through 4-4 contain information about each component. Tables 4-1 and 4-2 contain the software components for releases IR-1 and A, respectively, and Tables 4-3 and 4-4 contain the hardware components for releases IR-1 and A, respectively. The hardware components are characterized by Subsystem, CI, Service Group, Service Class, Subcomponents, and Location. The software components are organized by release build and thread, and are characterized by Subsystem, CI, Service Grouping, Component, Sub components, Type, SLOC, Development Track and Evaluation Package (EP), if applicable, and notable remarks. A legend containing the complete names of the builds, threads, and service groupings can be found after each table. A description of each component can be found in the CSMS Design Specification, 305-CD-003-002, and descriptions of the builds and threads can be found in the CSMS Integration and Test Plan, 319-CD-003-002.

Some components identified in Tables 4-1 and 4-2 show specific COTS-product selections. Such selections are not required for PDR (only make-vs.-buy decisions are due at this time); product selections, where indicated, are discussed in the CSMS Design Specification, 305-CD-003-002.

All hardware identified in Tables 4-3 and 4-4 is COTS.

Table 4-1. CSMS Components, IR-1 (1 of 2)

Build	Thread	Sub	CI	Svc Group	Component	Sub components	Type(s) ****	SLOC	Trk	EP	Remarks
Comm	Basic Dir. Svc	CSS	DCCI	Comm Fac.	File Access Service	FTP, KFTP, Custom FTP API	COTS, custom	2,000	I	EP5	Largely public domain and bundled software
Comm	Basic Interp	CSS	DCCI	Ob. Svcs	Message Passing	Synchronous	COTS, custom	1,000	I	EP5	
Comm	Dist. Time Svc	CSS	DCCI	Ob. Svcs	Time Service		COTS, custom	2,000	I	EP5	DCE UTS, with enhancements and glue code
Comm	Event Log	CSS	DCCI	Ob. Svcs	Event Logger Service		COTS, custom, hybrid	0	I	EP4	Reuse of APIs from EP4
Comm	Int. Msg	CSS	DCCI	Comm Fac.	Electronic Mail Service		COTS, custom	2,000	I	EP5	Mail packages bundled with OS, Custom API
Comm	Internet -wkg	CSS	DCCI	Comm Fac.	Virtual Terminal Service	Telnet, Ktelnet, X (for graphics)	COTS, custom	500	I	EP5	Public Domain
Comm	***	CSS	DCCI	Comm Fac.	Bulletin Board (BB)		COTS	0	I	EP3	Reuse BB from EPs
Comm	DCE Encap	CSS	DCCI	Distr. Ob Frwk	Dev. Support	OODCE	COTS, custom	2,500	I	EP5	I/F Definition Language & C++ Class Libraries
Comm	***	CSS	DCCI	Ob. Svcs	Thread Service		COTS, custom	500	I	EP5	COTS within DCE & OODCE
Comm	P to P	CSS	DCCI	Ob. Svcs	Directory/ Naming Service	BIND-DNS, CDS-DCE	COTS, custom	1,000	I	EP5	DCE implement. Upgrade in Rel-A.
Comm	Secure	CSS	DCCI	Ob. Svcs	Security Service	Authentication, Login	COTS, custom	1,500	I	EP5	Reuse of APIs from EP4
Comm	Secure	MSS	MCI	Mgmt App Svcs	Accountability Mgt: User Registration		COTS	0	I	n/a	
Comm	Secure	MSS	MCI	Mgmt App Svcs	Security Management for IR-1	Security Db Mgmt.	COTS, hybrid	500	I	n/a	DCE Authentication only
Internetwork	Internet -wkg	ISS	INCI	Network	Transport (T)	See Remarks	COTS	n/a	F	n/a	Embedded S/W within ISS H/W
Internetwork	Internet -wkg	ISS	INCI	Network	Network (N)	See Remarks	COTS	n/a	F	n/a	Embedded S/W within ISS H/W
Internetwork	Internet -wkg	ISS	INCI	Network	Data Link/Physical (D)	See Remarks	COTS	n/a	F	n/a	Embedded S/W within ISS H/W
Mgmt. Frwk	Alarm Pro & Dis	MSS	MACI	Com Mgmt Svcs	Management Agents	SNMP, Extensible	COTS, custom, hybrid	500	I	n/a	
Mgmt. Frwk	Alarm Pro & Dis	MSS	MCI	Com Mgmt Svcs	Maps & Collections	HP OpenView	COTS, hybrid	500	I	n/a	

Table 4-1. CSMS Components, IR-1 (2 of 2)

Build	Thread	Sub	CI	Svc Group	Component	Sub components	Type(s) ****	SLOC	Trk	EP	Remarks
Mgmt. Frwk	Alarm Pro & Dis	MSS	MCI	Com Mgmt Svcs	Mgmt. Framework (Monitoring, Discovery)	HP OpenView	COTS, hybrid	500	I	n/a	
Mgmt. Frwk	Alarm Pro & Dis	MSS	MCI	Mgmt App Svcs	Fault Management for IR-1		COTS, hybrid	500	I	n/a	Fault Identification, Fault Isolation
Mgmt. Frwk	Alarm Pro & Dis	MSS	MCI	Mgmt App Svcs	Performance Management for IR-1		COTS	0	I	n/a	
Mgmt. Frwk	Alarm Pro & Dis	MSS	MCI	Com Mgmt Svcs	Mgt User I/F	HP OpenView	COTS	0	I	n/a	
Mgmt. Frwk	Alarm Pro & Dis	MSS	MCI	Com Mgmt Svcs	DBMS	HP OpenView	COTS	0	I	n/a	
Mgmt. Frwk	Config. Mgmt	MSS	MLCI	Mgmt App Svcs	Configuration Management	ClearCase	COTS, hybrid	500	I	n/a	SW Config. Mgmt.
Mgmt. Frwk	***	MSS	MCI	Com Mgmt Svcs	Startup & Shutdown		hybrid	*	I	n/a	
Mgmt. Frwk	***	MSS	MCI	Com Mgmt Svcs	OA Tools		COTS	0	I	n/a	
Total Estimated SLOC development for IR-1								16k			

Legend:

Builds:

- Comm - Communications
- Mgmt. Frwk - Management Framework

Threads:

- Alarm Pro & Dis - Alarm Processing and Display
- Basic Dir. Svc - Basic Directory Services
- Basic Interp - Basic Interprocess Communications
- Config. Mgmt - Configuration Management S/W
- Dist. Time Svc - Distributed Time Service
- Event Log - Event Logging
- Int. Msg - Internet Messaging
- DCE Encap - DCE Encapsulation
- P to P - Process to Process Comms (RPC calls)
- Secure - Secure Logon/Logoff

Svc Group (Services Grouping):

- Com Mgmt Svcs - Common Management Services
- Mgmt App Svcs - Management Application Services
- Comm Fac. - Common Facilities
- Ob. Svcs - Object Services
- Distr. Ob Frwk - Distributed Object Framework

Trk:

Development Track

I - Incremental

F - Formal

* Included within the Management Software (MCI) development effort
*** Newly added components, not shown in the build/thread plan
**** See para. 3.1 for type definitions

Table 4-2. CSMS S/W Components, Release A (1 of 2)

Build	Thread	Sub	CI	Svc Group	Component	Sub components	Type(s) ****	SLOC	Trk	EP	Remarks
Comm Svcs	Email/ BB Svc	CSS	DCCI	Comm Fac.	Bulletin Board (BB)		COTS, custom, hybrid	1,000	I	EP6	Based on NNTP & public domain packages and IR-1 BB extensions
Comm Svcs	Email/ BB Svc	CSS	DCCI	Comm Fac.	Electronic Mail Service w/ MIME		COTS, custom, hybrid	500	I	EP6	Sending msgs to a Title
Comm Svcs	Dist. File Svc	CSS	DCCI	Comm Fac.	File Access Service	Remote File Access (DFS/AFS)	COTS, custom, hybrid	6,500	I	EP6	Remote File Access, custom schedule s/w
Comm Svcs	Comm Svcs	CSS	DCCI	Distr. Ob Frwk	Object Passing		COTS, custom	1,500	I	EP6	
Comm Svcs	Dir Name Svc	CSS	DCCI	Ob. Svcs	Directory/ Naming Service	GDS-X.500	COTS, custom	5,500	I	EP6	
Comm Svcs	Perf Mgmt	CSS	DCCI	Ob. Svcs	Event Logger Service		Custom	3,000	I	EP6	Class Library
Comm Svcs	Perf Mgmt	CSS	DCCI	Ob. Svcs	Event Service		COTS, custom	6,000	I	EP6	Candidate: Project Pilgrim
Comm Svcs	***	CSS	DCCI	Ob. Svcs	Life cycle Service		COTS, hybrid	4,000	I	EP6	Includes Startup & Recovery
Comm Svcs	Comm Svcs	CSS	DCCI	Ob. Svcs	Message Passing	Async, Deferred Sync	COTS, custom	8,000	I	EP6	DCE implementation
Internet work	Internet-wkg	ISS	INCI	Transport (T)	See Remarks		COTS	n/a	F	n/a	Embedded S/W within ISS H/W
Internet work	Internet-wkg	ISS	INCI	Network (N)	See Remarks		COTS	n/a	F	n/a	Embedded S/W within ISS H/W
Internet work	Internet-wkg	ISS	INCI	Data Link/ Physical (D)	See Remarks		COTS	n/a	F	n/a	Embedded S/W within ISS H/W
Mgmt. Svcs	Fault Mgmt	MSS	MCI	Mgmt App Svcs	Fault Mgt: Fault Recovery		Hybrid	500	F	n/a	CSS API I/F @ Rel-A.
Mgmt. Svcs	Fault Mgmt	MSS	MCI	Mgmt App Svcs	Fault Mgt: Reporting		Hybrid	500	F	n/a	HP OpenView, SQL, custom I/F @ Rel-A.
Mgmt. Svcs	Internet-wkg	MSS	MCI	Com Mgmt Svcs	Mgt Data Access		Custom , hybrid	4,000	F	n/a	
Mgmt. Svcs	Fault Mgmt	MSS	MACI	Com Mgmt Svcs	Management Agents	SNMP, Extensible	COTS, custom, hybrid	3,500	F	n/a	
Mgmt. Svcs	Internet-wkg	MSS	MCI	Com Mgmt Svcs	Maps & Collections	HP OpenView	COTS, hybrid	500	F	n/a	

Table 4-2. CSMS S/W Components, Release A (2 of 2)

Build	Thread	Sub	CI	Svc Group	Component	Sub components	Type(s) ****	SLOC	Trk	EP	Remarks
Mgmt. Svcs	Acctbilty	MSS	MCI	Mgmt App Svcs	Accountability Mgt: Accountability		COTS, custom	1,000	F	n/a	Accounting & Billing in Rel-B.
Mgmt. Svcs	Acctbilty	MSS	MCI	Mgmt App Svcs	Accountability Mgt: User Registration		COTS, custom	2,000	F	n/a	Accounting & Billing in Rel-B.
Mgmt. Svcs	Sys. Logistics Mgmt.	MSS	MCI	Mgmt App Svcs	Trouble Ticketing		COTS		F	n/a	
Mgmt. Svcs	Mgmt. Svcs	MSS	MCI	Mgmt App Svcs	Scheduling		COTS		F	n/a	
Mgmt. Svcs	Perf Mgmt	MSS	MCI	Mgmt App Svcs	Performance Mgt: Analysis		COTS, custom, hybrid	2000	F	n/a	Syst analy added @ Rel-A. Netwk only @ IR-1
Mgmt. Svcs	Perf Mgmt	MSS	MCI	Mgmt App Svcs	Performance Mgt: Monitoring		Custom	500	F	n/a	Syst monit added @ Rel-A. Netwk only @ IR-1
Mgmt. Svcs	Perf Mgmt	MSS	MCI	Mgmt App Svcs	Performance Mgt: Reporting		COTS, hybrid	500	F	n/a	
Mgmt. Svcs	Mgmt Frwk	MSS	MCI	Mgmt App Svcs	Management Framework		COTS, hybrid	1,000	F	n/a	
Mgmt. Svcs		MSS	MCI	Mgmt App Svcs	DBMS		COTS, custom, hybrid	2,000	F	n/a	
Mgmt. Svcs	Sys Logistic Mgmt	MSS	MLCI	Mgmt App Svcs	Configuration Management		COTS, hybrid	1,500	F	n/a	Baseline Mgmt., Change Request Mgmt.
Mgmt. Svcs	Sys Logistic Mgmt.	MSS	MLCI	Mgmt App Svcs	Physical Configuration Management		COTS		F	n/a	
Sys Sec	Sec Mgmt/ Acctbilty	CSS	DCCI	Ob. Svcs	Security Service	Remaining Features	COTS, custom	7,500 (6,000 for Sec)	I	EP6	Auth, Data Integ, Encrypt., Access Cont., User Reg.
Sys Sec	Sec Mgmt	MSS	MCI	Mgmt App Svcs	Security Mgt: Compliance Mgt		Hybrid	1,500	F	n/a	Conformity to policy.
Sys Sec	Acctbilty	MSS	MCI	Mgmt App Svcs	Security Mgt: Audit Info Collection		COTS	0	F	n/a	Logging of security data.
Sys Sec	Sec Mgmt	MSS	MCI	Mgmt App Svcs	Security Mgt: Intrusion Detection		COTS, custom, hybrid	1,500	F	n/a	Virus checkers. Addt'l types added @ Rel-A.
Sys Sec	Sec Mgmt	MSS	MCI	Mgmt App Svcs	Security Mgt: Reporting		COTS	0	F	n/a	
Total Estimated SLOC Development for Release A								66k			

Legend:

Builds:

- Comm Svcs - Communication Services
- Internetwork - Internetworking
- Mgmt Svcs - Management Services
- Sys Sec - System Security

Threads:

- Email/BB Svc - E-mail / Bulletin Board Services
- Dist. File Svc - Distributed Files Services
- Comm Svcs - Communications Services
- Internet work - Internetworking
- Fault Mgmt - Fault Management
- Acctbilty - Accountability Management
- Perf Mgmt - Performance Management
- Dir Name Svc - Directory/Naming Service
- Sys Logistic Mgmt - System Logistics Management
- Sec Mgmt - Security Management

Trk:

Development Track

I - Incremental

F - Formal

Legend: (cont.)

Service Groupings:

- Comm Fac. - Common Facilities
- Distr. Ob Frwk - Distributed Object Framework
- Ob. Svcs - Object Services
- Com Mgmt Svcs - Common Management Services
- Mgmt App Svcs - Management Application Services

*** Newly added components not shown in build/thread plan

Table 4-3. CSMS Deployed Hardware Components, IR-1

Sub	CI	Service Group	Service class	Sub components	Location(s)
CSS	DCHCI	Distr. Comp H/W	Medium File Server	Enterprise Comm Server	EDF (moved to GSFC/SMC in Rel. A)
CSS	DCHCI	Distr. Comp H/W	Medium File Server	Bulletin Board Server	EDF
CSS	DCHCI	Distr. Comp H/W	Printer	N/A	EDF, GSFC, EDC, LaRC, MSFC
MSS	MHCI	Mgmt. Sub H/W	Small File Server	Local Mgmt Server	GSFC, EDC, LaRC, MSFC
MSS	MHCI	Mgmt. Sub H/W	Medium File Server	Enter. Mgmt Server	EDF (moved to GSFC/SMC in Rel. A)
MSS	MHCI	Mgmt. Sub H/W	Small Workstation	Mgmt Workstation	2 @ EDF
ISS	INHCI	Network H/W	n/a	Low-end Ethernet Hub	EDC
ISS	INHCI	Network H/W	n/a	Ethernet Cabling	EDC

Note: Since IR-1, as deployed, reuses the V0 LAN and WAN, a limited set of ISS components are shown.

Legend:

Service Group:

- Distr. Comp H/W - Distributed Computing Hardware
- Mgmt. Sub H/W - Management Subsystem Hardware
- Network H/W - Network Subsystem Hardware

Table 4-4. CSMS Deployed Hardware Components, Release A

Sub	CI	Service Group	Service class	Sub components	Location(s)
CSS	DCHCI	Distr. Comp H/W	Small File Server	Local Comm Server	GSFC, GSFC-EOC, EDC, LaRC, MSFC
CSS	DCHCI	Distr. Comp H/W	Printer	N/A	GSFC-EOC
ISS	INHCI	Network H/W	n/a	FDDI Switch	GSFC, LaRC, MSFC
ISS	INHCI	Network H/W	n/a	FDDI Concentrator	GSFC, LaRC, MSFC
ISS	INHCI	Network H/W	n/a	FDDI Cabling	GSFC, LaRC, MSFC, SMC, GSFC-EOC
ISS	INHCI	Network H/W	n/a	Ethernet Cabling	GSFC, LaRC, MSFC, EDC, GSFC-EOC
ISS	INHCI	Network H/W	n/a	Router Interface (WAN)	GSFC, LaRC, MSFC, EDC, GSFC-EOC
ISS	INHCI	Network H/W	n/a	Test Equipment	GSFC, LaRC, MSFC, EDC, GSFC-EOC
MSS	MHCI	Mgmt. Sub H/W	Small File Server	Local Mgmt Server	GSFC-EOC
MSS	MHCI	Mgmt. Sub H/W	Small Workstation	Mgmt Workstation	2 each @ GSFC, GSFC-EOC, EDC, LaRC, MSFC
ISS	INHCI	Network H/W	n/a	FDDI/Ethernet Hub	EOC

Legend:

Service Group:

Distr. Comp H/W - Distributed Computing Hardware
Mgmt. Sub H/W - Management Subsystem Hardware
Network H/W - Network Subsystem Hardware

4.2 CSMS Development Estimation

The development effort for CSMS is broken into two primary pieces; detailed design and component development. As already discussed, CSMS components are composed of hardware, stand-alone COTS software, custom and/or hybrid software or a combination of two or more software products. Table 4-5 provides examples of the type of effort planned for each software component during the detailed design, and code and unit test phases of development. Although categorized as a code and unit test activity, effort to integrate COTS products, and reused software, with custom and hybrid code is also included in our estimates. This activity is at the component level and should not be confused with the integration of unit-tested components into functional threads and builds.

Table 4-5. CSMS Development Activities

	Custom	Hybrid	COTS-based
Detailed Design	• API definition • PDL • Completion of dynamic models and event traces	• Allocation of low-level functions to script definition • Screen definition • Report format definition • IDL definition	• DBMS schema completion • Final definition of management data collection from COTS MIBs • Detailed device configuration definition (routers, hubs, hosts) • Detailed DCE cell configuration • Identification of site specific parameters/settings for COTS s/w installation
Code & Unit Test	• C++ code development; compile, link, run, unit test	• Develop & test Unix scripts, GUI-builder scripts, 4GL files • Generate and test customized MIBs with ECS applications. • Generate and test IDL with specific client/server applications	• Generation of DBMS • Installation of COTS s/w and test of specific functions • installation of h/w, generation of device configuration files (e.g., routing tables) • DCE cell deployment / installation

4.2.1 Custom Software Estimation Modeling

The software development estimation effort began by comparing the CSMS functional and performance requirements to historical data from past programs to estimate the lines of source code. These line-of-code counts have been input to REVIC (Revised Intermediate COCOMO), a software estimation tool based on the COCOMO model, developed by Dr. Barry W. Boehm, in his book *Software Engineering Economics*. REVIC was developed by Hughes under a DoD contract, and its use is now authorized by all government agencies and their contractors. REVIC uses a set of 19 user defined variables to influence the estimated development time and man loading by characterizing the development process established for ECS. These variables consider the use of structured analysis and OO design methodologies thus offering better predictions than previous versions. REVIC also considers the size of the modules in establishing coding efficiencies. The model runs in two distinct passes. First it predicts the labor estimates for the entire development life cycle, expressed as the number of staff months required for each LOC count. Next this effort is broken down to the major development phases including requirements

analysis, preliminary design, detailed design, code and unit test, integration & test, and development test and eval. For this development plan, we have used the estimates for detailed design and software code and unit test, to validate the feasibility of meeting the established schedule milestones.

4.2.2 Hybrid Software Development

One of the major concerns regarding the line-of-code counts presented at PDR was whether the estimates included scripting and 4GL type of code development, and to what degree this "hybrid software" contributes to the estimated development schedule and staffing profile. In response to this concern, the CSMS staff has reviewed the LOC estimates, and compared the effort associated with development of hybrid software with that for C/C++ software. Several key points have driven our decision to continue to use one software estimation model for both types of code. These include a comparison of development tools, ease of learning, and size of experienced developer base between these two types of coding processes.

We first considered the development environment of each code type to see if one facilitated greater productivity than the other. We found that the development of custom software was more efficient than development of hybrid software, due to the wealth of integrated productivity tools available for the high-level programming languages, including syntax checkers.

Custom development toolsets provide instant feedback to the developer, and provide on-line remedies for coding errors. By the time a custom software module is run, most, if not all of the syntax errors have been resolved thus dramatically reducing development time. Hybrid software is typically developed in an interpretive environment using the "trial and error" paradigm, and the only feedback is from the runtime outputs. The hybrid code developer must rework his code until the expected results are achieved. This lack of support tools limits the efficiency of hybrid software development and provides the rationale for estimating more effort per line of hybrid code than for custom code development.

A second major point, that is, learning curve, offsets the development tool argument. HLLs offer greater functional capability than the scripting and 4GL languages used to develop hybrid software. However, this greater capability comes with a larger and more complex syntax requiring a longer learning curve than that of a typical scripting language. The hybrid software languages typically provide a limited syntax designed around the command set of its parent application. These languages are generally more intuitive to use, and with a smaller syntax, result in shorter learning curves than that of the HLL. This second consideration offsets the first argument by estimating less effort per line of hybrid code than that of custom code. Based on these two arguments we have concluded that the original software estimation model is suitable for both custom and hybrid software development planning for CSMS.

4.2.3 COTS-based Development

COTS-based development has been identified as a major element of the CSMS development plan due to our extensive use of COTS software. COTS-based development refers to the activity associated with configuring and tuning the installed products, and integrating COTS products, with associated custom and/or hybrid software elements for each end-item component. This also

includes the use of previously developed software, since it will need to be installed and tested on the selected hardware platforms to ensure compatibility with co-resident software.

Since PDR, we have generically identified (and specifically, in a few cases) all of the COTS products required for IR-1 and for Release A. Our current estimates include the COTS-based detailed design and "code and unit test" activities presented in Table 4-5. The estimates for each release are presented in their respect development section. The primary factors considered in these estimates include the degree of COTS complexity (e.g. number of functions, software size, client/server based), number of different platforms and site unique configurations, and staff experience with each product. A more detailed explanation of the methodology used to generate the COTS development estimates, including a breakdown of the effort by component is presented in Appendix A.

The primary purpose of these estimates are to ensure that ample development time and resources have been planned to meet the contractual delivery schedule of each release. The aggregation of components to the thread level is addressed in the segment Integration and Test Plan for each release.

5. IR-1 Development Schedules

5.1 IR-1 Objectives

The objectives of Interim Release 1 (IR-1) are to provide ECS components to support early testing of the TRMM interfaces, and provide support for TRMM and EOS-AM-1 Algorithm Integration and Test. The CSMS IR-1 components provide a communications infrastructure and limited system management functionality to aid in testing the TRMM ground system and support the integration of science software into the IR-1 DAACs.

Section 4 of this document identifies the CSMS components to be developed for IR-1. This section describes the overall IR-1 development process and explains how the components are grouped to support the functional threads identified in the CSMS Integration and Test Plan.

5.2 IR-1 Development

The development plan for IR-1 begins with the completion of PDR and includes the detailed design, code and unit test, and integration and test efforts required to complete the CSMS IR-1 builds. Also included is the EDF configuration effort, and migration paths for integration of previously developed components such as the EP3 Bulletin Board Service and selected prototypes. This plan, as illustrated in Figure 5-1, *Interim Release-1 Development Plan*, allows for 8 calendar months to complete the segment builds. This is further broken down into 5 months for detailed design through code and unit test, and 3 months for build/thread integration and test. Our resource estimates, presented in Table 5-1, *Development Estimates for Interim Release 1* have determined that 97.5 staff months will be required to develop the software components of IR-1. Details of the COTS development estimates can be found in Appendix A, *Basis of Estimate for Development Effort* . Considerable overlap has been planned amongst the activities to ensure that the communications infrastructure and management framework portions will be completed early in order to support the applications which rely on them. The detailed planning process further delineates the allocation of staff to the various activities.

As described in Section 3.2, the development schedule is documented and maintained in the IR-1 portion of the ECS Intermediate Logic Network (ILN), CDRL #008, 108-CD-001-XXX. The ILN schedule is used to prepare the detailed development schedules and work packages for managing the development process. The detailed schedules, prepared by the cost account (development) managers, offer a more detailed view of the ILN activities and provide the level of detail needed to properly evaluate progress, and identify problems before they cause major impacts to the delivery schedule or other dependent tasks.

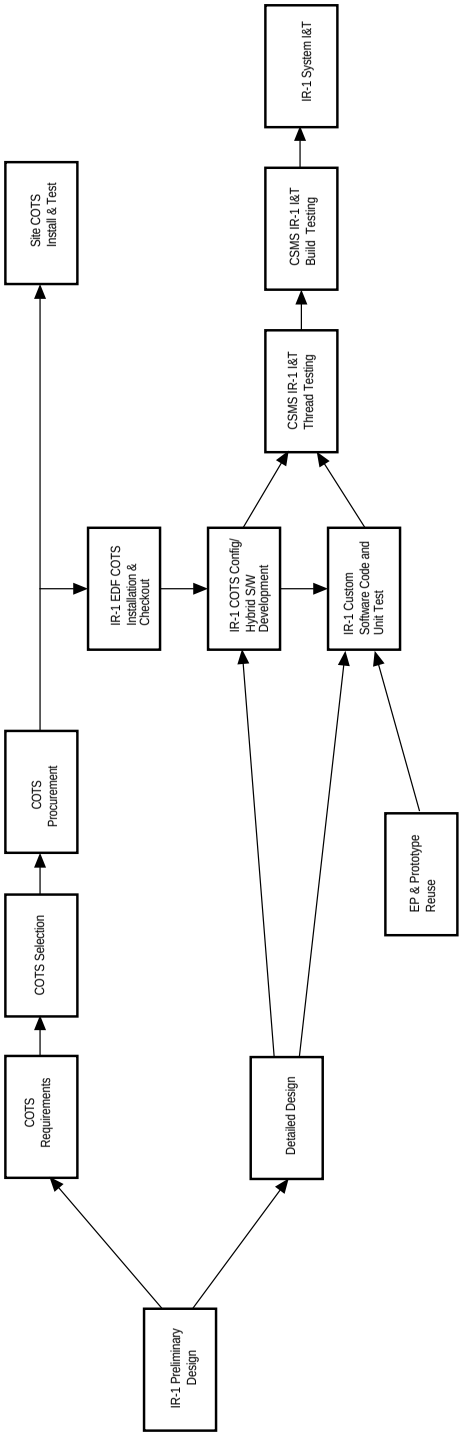

Figure 5-1. CSMS Interim Release 1 Development Plan

5.2.1 IR-1 Detailed Design

The detailed design phase of the IR-1 release began with the completion of the CSMS Preliminary Design Review (PDR) for IR-1/Release A. All incrementally developed software used for IR-1 will be developed concurrent with the detailed deign effort leading to formal development. During this phase, the preliminary design presented in the PDR documentation package, and at the CSMS PDR is further refined into a stable, well-defined detailed design based on requirements allocations and architecture from PDR.

The IR-1 detailed design will include COTS software and hardware selections, including vendor, model, and part numbers for all proposed IR-1 products. COTS configuration designs will be completed and documented. Power budgets and siting plans, cabling runs, and demarcation points will also be defined.

5.2.2 COTS Requirements

Once the segment design team has developed and validated a logical design, they will prepareproduct specifications based on the requirements developed during the logical design (Level 4 requirements). In addition to preparing these specifications, a justification for each requirement/specification is also furnished to ensure traceability back to the logical design. By providing justification for each requirement, any "extras" in the design are eliminated. It is important that these specifications are complete, yet do not suggest a vendor specific solution. The competition is open to all vendors who can meet the stated requirements and specifications.

5.2.3 COTS Selection

Based upon the detailed requirements specifications developed by the segment design teams, EDS will prepare an RFI/RFC that is released to the government and vendor communities for review and comment. An RFC will always be issued for items of high value, or those which are technically complex or entail highly specialized procurements. The RFC allows the vendor to respond with questions, and ask for clarification. The RFC informs the vendor that a RFP is forthcoming. The RFC also contains draft specifications necessary for prospective vendors to prepare their proposals. By responding to vendor questions and clarifying issues, it is anticipated that when the solicitation is released the vendor community will respond with clear, concise proposals. After review of the RFC comments and development of evaluation criteria, an RFP package is assembled by EDS and routed through the ECS community prior to formal release to NASA for consent to release. Once approved the RFP will be released to the vendor community.

Once the proposals are submitted to EDS by the vendor, costing will be evaluated separately from the technical requirements. Upon receipt of the proposal, EDS will perform a preliminary validation. Proposals meeting minimum mandatory requirements will then be further evaluated to determine and score the technical and cost proposals. The costing evaluation will be conducted by the EDS Procurement Management team with assistance from the technical evaluation team. Technical evaluators will not review the cost proposals until after scoring of the technical proposals is complete. The cost evaluation will occur at the time of the technical evaluation and generally after a proposal has met minimum mandatory requirements. This evaluation verifies that the

submitted pricing includes all mandatory specifications and that all contractual issues have been met.

The Technical Evaluation Team will receive technical proposals from the EDS procurement management team and will score each submission based on its technical merits. Factors such as vendor stability and financial position, vendor support, product training, and documentation are included in the evaluation criteria.

Using the weight factors assigned, the technical scores are recorded. When the vendor offers a feature that is deemed a benefit to the program, the extra value may be converted into dollars. This additional value will be incorporated into the total life cycle cost.

A combined cost and technical team will then determine a "Best Value" ranking. "Best Value" is the sum of technical score, price rankings, and the "extra" capability provided by each vendor. "Best Value" is the basis for contract awards.

Benchmarking and stress testing will not be performed for all products. EDS will require vendors to certify their compliance with applicable federal and ECS standards, appropriately including witnessed runs of compliance suites. In addition, vendor claims are evaluated by the Hughes/EDS team through prototyping in the science and technology laboratory (STL).

EDS, upon completion of the "Best Value" evaluation will prepare a Source Evaluation Recommendation report. This report will provide an overview of the requirement and the segment(s) that the acquisition will support. The report will also provide an overview of the technical and price evaluation and the recommendation to purchase. After review by the ECS team, the Source Evaluation Recommendation as well as the negotiated subcontract are delivered to NASA's Contracting Officer for review, approval, and consent to issue.

5.2.4 COTS Procurement

EDS will issue a purchase order to the selected vendor. The purchase order is generated and tracked via the corporate-based, but project-controlled, EDS contractor purchasing system (CPS). This automated procurement system was approved by the Defense Contract Management Area Operations-Philadelphia (DCMAO-Philadelphia) on 1 March 1993. A key internal feature of CPS is the procurement and inventory control system (PICS), which provides on-line status of each purchase order. Additionally, the procurement process will be recorded and tracked via the VCATS database. EDS will ensure that vendors respond to the schedule requirements that are stated in the purchase order.

When a solicitation is awarded or a purchase order has been issued, a contract file will be kept to include historical background on the purchase action. This file will provide an audit trail as well as a useful tool for analysis of past trends and results. The contract file will show the description of services, type of solicitation, a well-defined statement of work, delivery schedules, shipping terms, invoice and payment terms, flowdown clauses, amount or estimate of purchase, description of evaluation criteria, and persons to contact if questions arise. The contract file will demonstrate through documentation that all purchasing actions have been made in accordance with acceptable government procedures, as well as practicing fair and open competition.

5.2.5 COTS Installation and Checkout

As alluded to in Section 4.2, all COTS products procured for development will be subjected to the NASA-approved inventory control process to verify that all COTS products received are as expected and in acceptable condition for use on ECS. Hardware products will undergo a burn-in test using the manufacturer's recommended test procedures. Once the hardware has been accepted, it will be configured with any standard operating and utility software and configured to the minimum manufacturer-recommended operating state. Finally, COTS application software will be installed and the machine will be considered ready for development use.

5.2.6 EP & Prototype Reuse

Some of the software developed for Evaluation Packages and prototypes will be reused to satisfy ECS functional and performance requirements. This software may need to be modified, require documentation upgrades, or require more intense testing than was originally performed. Software developed for EP use will have undergone testing to the same level as that for incrementally-developed software, hence it may only need regression testing to verify compatibility with newly developed software. Software developed for prototyping activities, is typically produced at a more rapid pace, and much of the checking and testing is bypassed to enable it to be fielded sooner than traditionally developed code. Regardless of development source, all previously developed software must be evaluated to determine the amount of modification, testing, and documentation cleanup needed to bring it up to established standards prior to integration with IR1. Enhancements to existing software may be made prior to or during integration depending on the nature of the changes, level of existing documentation, and testing requirements.

5.2.7 IR-1 Custom Software Code and Unit Test

The custom software code and unit test task represents the effort required to produce and test software modules developed using high-level programming languages such as C++. This effort, as explained earlier, is characterized in terms of calendar months, and the number of software developers required per task. While this effort is presented as a single task, it is actually composed of all of the pieces of custom software required for IR-1. These tasks transform the Program Design Language (PDL) documentation, and detailed design specifications prepared during the detailed design phase into standalone software modules which are unit tested. Next these modules are further integrated with other modules/COTS products, or are handed off to the Segment Integration and Test organization where they are placed under Configuration Control and prepared for formal test.

5.2.8 IR-1 COTS Software Configuration and Hybrid Software Development

A substantial portion of the CSMS development effort includes development of the hybrid software modules, as described in Sections 3 and 4, used to tailor COTS products such as the UNIX operating system, the Configuration Management Tool, and the Management Framework product to satisfy the operational, functional and performance requirements assigned to the CSMS for IR-1. In addition to this development activity, the COTS products need to be configured and tuned to the operating environment, and in many cases they need to be integrated with other COTS products in order to fulfill higher level functions. Since PDR we have analyzed the COTS products identified for IR-1, those already selected, and candidate products for the remaining slots

to ensure that adequate time and resources will be allocated to support this effort. Since the last release of this document, we have added this COTS development labor to the overall development estimates presented in Table 5-1, *Development Estimate for Interim Release 1*. We have also included a detailed breakdown of the COTS development labor by component in Appendix A, *Basis of Estimate for Development Effort*.

5.2.9 CSMS IR-1 I & T Development and Test

Once the IR-1 software components have completed unit test, they are handed off to the Segment Integration and Test organization where they are combined with other components, or builds to begin functional thread testing. As individual functional threads are completed, they are combined with other threads and sub-builds until the two primary builds are completed. These builds are turned over to the System Integration and Test organization for integration with the builds from the other segments. Further details of the build development activities are provided in the CSMS Integration and Test Plan for the ECS Project Volume 1: IR-1.

5.3 IR-1 Components Mapping to Threads and Builds

The build/thread portion of the CSMS Integration and Test Plan for the ECS Project, Volume 1: IR-1 is a major influence for sequencing the development of the IR-1 components. The build/thread plan includes two primary builds, the Communications build, and the Management Framework build. These two builds provide all of the CSMS functionality required for IR-1. In addition, are two secondary builds, the Internetworking build and the OSF-DCE build which contain an aggregation of networking and distributing computing functional threads. The two secondary builds are integrated with system management and communication functional threads, to form the Communications and Management Framework builds. The time phasing of these builds and their underlying functional threads drive the development schedule for the IR-1 components.

In Section 4, we presented a breakdown of the CSMS subsystems to components, sorted by functional thread and/or build. In this section we have taken the SLOC estimates passed through the REVIC model to arrive at our schedule and staffing requirements for the software development effort. Additionally, we have identified some 20 COTS products (some bundled within others) and have prepared estimates for integration of these products with developed software to produce the components identified in Table 4-1. As described in Section 4.2.3, the COTS development effort for CSMS is remarkable to the extent that a product by product analysis for this effort was performed. Each product or product category was reviewed and the total effort is presented in Table 5-1, *Development Estimates for Interim Release 1*. Details of the analysis are presented in Appendix A. Due to the large amount of overlap between the COTS products, with respect to the functional thread distribution, we have chosen to present the COTS development effort in terms of staff months for detailed design effort and for "code and unit test" effort for the entire release. One of the goals of this document has been to offer guidance by which the cost account managers can perform the detailed planning for development of the CSMS components. Figure 5-2, *Component Mapping to the Build/Thread Plan*, illustrates, at a high level, the format of the resource loaded development schedule planned for IR-1.

Table 5-1. CSMS Development Estimates for Interim Release 1

	SLOC	Minimum Schedule (Calendar Months)			Staff Months		
		Design	CUT	Total	Design	CUT	Total
Custom/Hybrid Software Development							
IR-1 Basic Dir. Services	2000	2.0	1.2	3.3	2.04	1.55	3.59
IR-1 Basic Interprocess Communication	1000	1.6	0.9	2.5	0.89	0.67	1.56
IR-1 Distributed Time Service	2000	2.0	1.2	3.3	2.04	1.55	3.59
IR-1 Event Logging Thread	0	0.5	0.3	0.8	1.00	1.00	2.00
IR-1 Internet Messaging	2000	2.0	1.2	3.3	2.04	1.55	3.59
IR-1 Internetworking Sub-build	500	1.2	0.7	1.9	0.39	0.29	0.68
IR-1 Process to Process Communications	1000	1.6	0.9	2.5	0.89	0.67	1.56
IR-1 Secure Logon/Logoff	2000	2.0	1.2	3.3	2.04	1.55	3.59
IR-1 Alarm Processing and Display	2000	2.0	1.2	3.3	2.04	1.55	3.59
IR-1 Communications Build	500	1.2	0.7	1.9	0.39	0.29	0.68
IR-1 Configuration Management	500	1.2	0.7	1.9	0.39	0.29	0.68
IR-1 Management Framework Build	0	0.5	0.75	1.3	1.00	1.00	2.00
IR-1 DCE Encapsulation Thread	2500	2.2	1.3	3.5	2.52	1.91	4.42
COTS Development Effort					49.50	16.5	66.00
Totals	16000						97.53

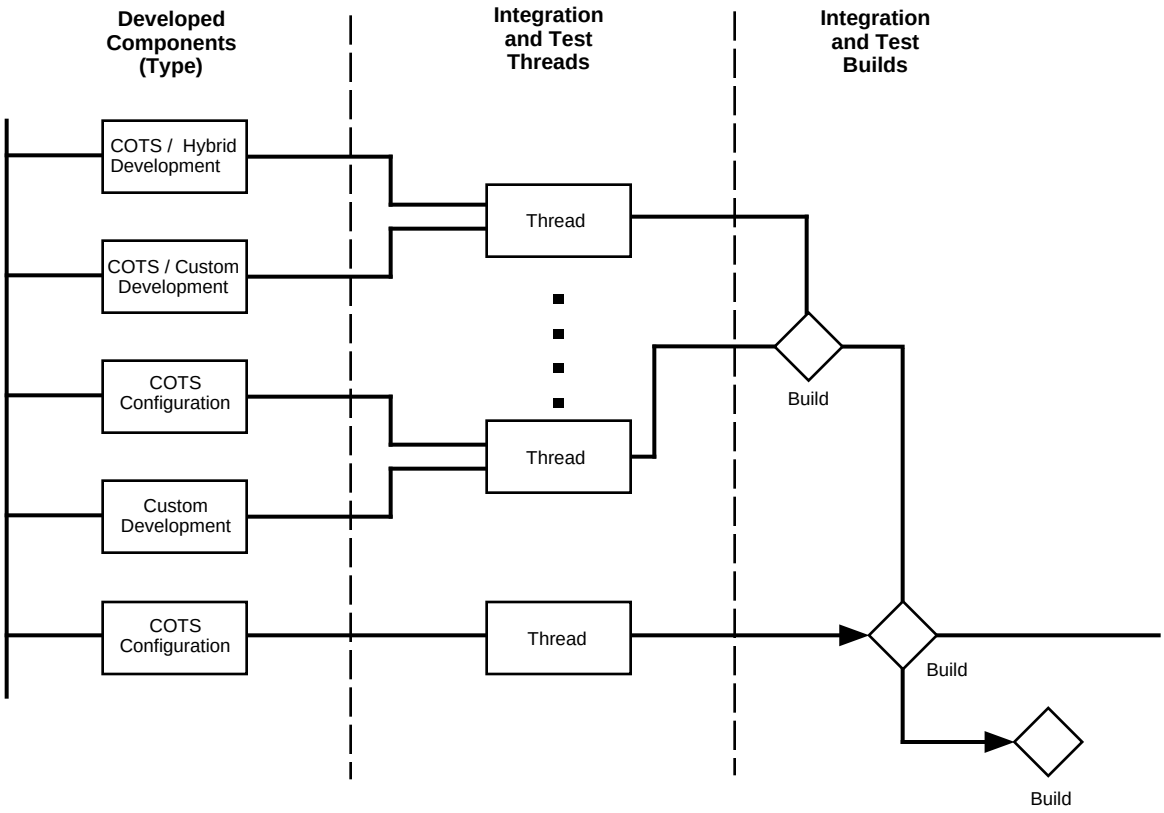

Figure 5-2. Component Mapping to the Build/Thread Plan

6. Release A Development Schedules

6.1 Release A Objectives

The objectives of ECS Release A are to provide additional functionality to support the TRMM mission, and SDPS mission and user services: Version 0 data migration, EOS-AM-1 and Landsat-7 interface testing, and EOS-AM-1 algorithm integration and test. Details of the Release A functionality can be found in the CSMS Design Specification for the ECS Project, 305-CD-003-002.

Section 4 of this document identifies the CSMS components to be developed for Release A. This section describes the overall Release A development process and explains how the components are grouped to support the functional threads identified in the CSMS Integration and Test Plan, Volume 2: Release A, 319-CD-004-002.

6.2 Release A Development

The development plan for Release A begins with the completion of PDR and includes the detailed design, code and unit test, COTS requirements definition and procurement, and integration and test efforts required to complete the CSMS Release A builds. CSMS components for Release A are developed both formally, and incrementally, as described in the Multi-track Development White Paper, 194-WP-904-002. The Release A Critical Design Review is currently planned for August 1995. By CDR, all detailed design activities are expected to be complete and the code and unit test activities for the well defined components will be underway. This plan, as illustrated in Figure 6-1, *CSMS Release A Development Plan* allows for 13 calendar months to complete the segment builds. This is further broken down into 10 months for detailed design through code and unit test, and 3 months for build/thread integration and test. Our resource estimates, presented in Table 6-1, *Development Estimates for Release A* indicate that 280 staff months will be required to complete the detailed design, and code and unit test activities for the CSMS Release A components. Details of our design approach and software development efforts are included in the following sections. Appencix A, *Basis of Estimate for Development Effort,* contains details of the COTS development effort by Release A component. Details of the remaining schedule elements are similar to IR-1 activities of the same name and can be found in Section 5.

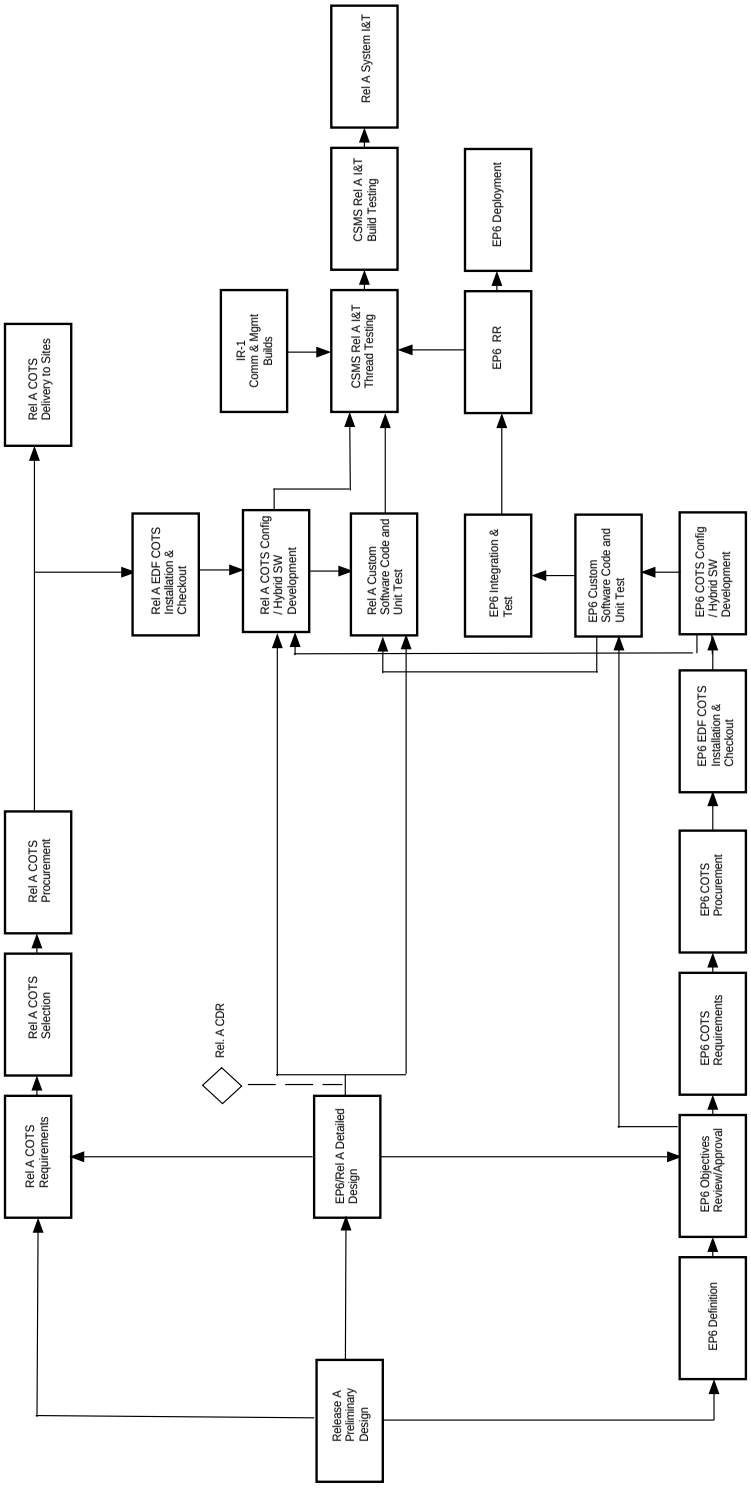

Figure 6-1. CSMS Release A Development Plan

6.2.1 Detailed Design

The detailed design activities for Release A are expected to begin at the completion of the CSMS PDR. During this phase, the preliminary design presented in the PDR documentation package, and at the PDR is further refined into a stable, well-defined detailed design based on the requirements allocations and architecture established at PDR. EP6 objectives, which are based on feedback from previous EPs, early IR-1 feedback, and Release A requirements allocated to the incremental track will be established and approved at the beginning of the detailed design phase. The EP6 objectives review also initiates the EP6 COTS procurement cycle, which is critical to the Release A development since the code and unit test and COTS development efforts cannot begin without the selected COTS products in house and configured for use.

The Release A detailed design will include COTS hardware selections, including vendor, model, and part numbers for all proposed Release A hardware. In addition, power budgets and siting plans, cabling runs, and demarcation points will be defined.

Detailed software designs will be completed for each component, characterized as: Custom software, COTS software, or Hybrid software.

Custom software will consist of C/C++ code modules of typically 250 lines or less. They will be described in object-oriented detailed design models, Program Design Language (PDL) listings, and API module calling sequences/semantic constructs, as appropriate.

COTS software selections will be made using traditional engineering trade studies for operating systems, system management and administration applications, network management and configuration tools, various utilities, software and script development tools, DBMS, libraries, agents and management information bases, MIBs, etc. Configuration tables and other specifics associated with each of the selected products will be prepared.

Hybrid software solutions will be presented based on: traditional engineering studies as described for the COTS software selections, and rationale and design data for hybrid solutions. Detailed descriptions of the scripting efforts including 4GL and UNIX shell scripts; display definitions and Motif templates; MIB tailoring data, including variable definitions; ECS-unique configuration solutions for software packages such as HP OpenView and ClearCase; detailed schema for DBMSs; and router address, security and other configuration table definitions.

6.2.2 COTS Requirements

The COTS Requirements task for Release A follows the same procedure as that developed for Ir1, which is described in Section 5.2.2.

6.2.3 COTS Selection

The COTS Selection process for Release A is the same as that used for Ir1, which is described in Section 5.2.3.

6.2.4 COTS Procurement

The COTS Procurement activities for Release A follows the same procedures as that developed for Ir1, which is fully described in Section 5.2.4.

6.2.5 Release A COTS Configuration/Hybrid Software Development

The COTS configuration/hybrid software development effort for Release A will be similar to that planned for IR-1, although we expect to benefit from the lessons learned during the IR-1 development in terms of streamlined processes, and shorter learning curves. Once the Release A hybrid software modules have been incorporated with their associated COTS software, they are merged with custom software modules, or submitted to the I & T organization for build/thread testing.

6.2.6 Release A Custom Software Code and Unit Test

The development of custom software for Release A begins just prior to CDR with the incrementally developed software for EP6. Release A formal track code and unit test begins after CDR, paralleling EP6 development. As mentioned in Section 5, special attention was given to analyzing the COTS development effort on ECS since over 40 COTS products (some bundled within others) are planned for use in the Release A components of the CSMS. The incorporation of COTS software with custom software will be performed once the hybrid software development for a given component has been completed. Specific details of the software development methodologies, and standards can be found in the Software Development Plan for the ECS Project, DID 308.

6.2.7 Release A Integration and Test

Once all of the custom software, hybrid software, and COTS products composing a software component have been combined and unit tested, it is submitted to the integration and test organization for formal testing. Details of the integration and test plans can be found in the CSMS Integration and Test Plan for the ECS Project, Volume 2: Release A, 319-004-002.

6.3 Release A Components Mapping to Threads and Builds

The build/thread portion of the CSMS Integration and Test Plan for the ECS Project, Volume 2: Release A is a major influence for sequencing the development activities of the Release A components. This sequence was established to ensure that the Release A delivery of CSMS will support the delivery of all ECS subsystems by contract milestone dates established for Release A. The CSMS build/thread plan for Release A establishes three builds, the Communication Services build, the Management Services build, and the System Security build. When combined, these builds provide all of the CSMS functionality required for Release A. In addition to the newly developed components for Release A, the IR-1 Management Framework build and Communications builds are integrated into the Release A builds. The time phasing of these builds and their underlying functional threads are the major drivers of the Release A development schedule, although activities such as COTS procurement, prototype development, and product reuse will also influence the final plan. As described in Section 4.2.3, the COTS development effort for CSMS is remarkable to the extent that a product by product analysis for this effort was

performed. Each of the 40 some COTS products or product categories was reviewed. Due to the large amount of overlap between the COTS products, with respect to the functional thread distribution, we have chosen to present the COTS development effort as the number of staff months for detailed design and for "code and unit test" of the Release A components of the CSMS. The results are provided in Table 6-1, *Development Estimates for Release A.* Details of the COTS development estimates are provided in Appendix A.

Table 6-1. Development Estimates for Release A

	SLOC	Minimum Schedule (Calendar Months)		Staff Months		
		Design	CUT	Design	CUT	Total
Custom/Hybrid Software Development						
Release A Email/Bulletin Board	1500	1.8	1.1	1.4	1.1	2.54
Release A Distributed File Service (DFS)	6500	3.2	1.9	8.4	6.4	14.78
Release A Directory/Naming Service	5500	3.0	1.8	6.9	5.2	12.09
Release A Communication Services Build	13500	4.3	2.6	20.2	15.3	35.52
Release A Performance Management Thread	12000	4.1	2.4	17.5	13.3	30.84
Release A Fault Management Thread	4500	2.8	1.7	5.4	4.1	9.51
Release A Internetworking Thread	4500	2.8	1.7	5.4	4.1	9.51
Release A Accountability Thread	7500	3.4	2.0	10.0	7.6	17.55
Release A Security Management Thread	9000	3.6	2.2	12.4	9.4	21.84
Release A System Logistics Management Thread	1500	1.8	1.1	1.4	1.0	2.40
COTS Development Effort				92.25	30.75	123.00
Total	66000					279.58

This page intentionally left blank.

7. Release B Development Schedules

This section will be supplied with the Release B submittal due at Release B IDR.

This page intentionally left blank.

8. Release C Development Schedules

This section will be supplied with the Release C submittal due at Release C IDR.

This page intentionally left blank.

9. Release D Development Schedules

This section will be supplied with the Release D submittal due at Release D IDR.

This page intentionally left blank.

10. Product Delivery

10.1 General

This section of the Release and Development Plan was originally intended to provide traceability of the product builds that composed each delivery. A separate section was established since it was envisioned that components developed by other segments might offer functionality required within CSMS and therefore would not need to be developed twice. In this case, the traceability of components outside of CSMS would be documented herein.

10.2 CSMS Products at IR-1 and Release A

The first formal release of this document is in support of the Interim Release 1 and Release A missions. As of the document submittal date, the use of components developed outside of the CSMS in support of IR-1 or Release A is not planned. This being the case, the traceability matrix planned for this section has already been provided in Tables 4-1 and 4-2.

This page intentionally left blank.

Appendix A: Basis of Estimate for Development Effort

The following table provides the estimates for development of each CSMS component by release. The table was originally developed prior to the ECS system design review in June 1994. Since then it has been updated to reflect the current CSMS design. The line-of-code estimates shown are the same as that in Tables 4-1 and 4-2 of this document and are included for reference only. What is new are the estimates to transform COTS products from an "out of the carton" state to an ECS ready state where it performs the functionality for which it was procured.

The table is organized by CSMS subsystem, CI, and component. Each component is described by development type, that is, a COTS product, a custom software product, or a Hybrid software product, or a combination of two or more, as previously defined in Section 4.

The development estimates presented herein have been established using the same method that was described and used in the technical proposal for the ECS program. Those currently selected and/or candidate COTS products were evaluated and rated in terms of difficulty. Three levels of difficulty were established: minimal, average, or complex. Those products considered to require a minimal amount of difficulty were allocated one staff month (SM) for development; products rated as average were allocated two staff months, and those products considered complex were allocated four months. For extremely large COTS products such as OODCE, a further breakdown was made in order to provide more realistic estimates. The following table contains the results of the evaluation in terms of staff months for each component.

Subsystem	CI	Component	Dev. Type	Ir1		Rel. A		Remarks
				SLOC	SM	SLOC	SM	
CSS	DCCI	File Access Service	COTS/Custom	2,000	2	6,500	4	
CSS	DCCI	Message Passing	COTS/Custom	1,000	2	8,000	4	Asyncrhonous messages
CSS	DCCI	Message Passing	COTS/Custom				2	Deferred synchronous
CSS	DCCI	Time Service	COTS/Custom	2,000	4	n/a	4	
CSS	DCCI	Event Logger Service	COTS/Custom/Hybrid			3,000	4	
CSS	DCCI	Electronic Mail Service	COTS/Custom/Hybrid	2,000	2	500	2	
CSS	DCCI	Virtual Terminal Service	COTS/Custom	500	4	n/a		
CSS	DCCI	Bulletin Board Service	COTS			1,000	4	
CSS	DCCI	Developer Support	COTS/Custom	2,500	4	n/a	4	OODCE - IDL
CSS	DCCI	Developer Support	COTS/Custom				4	OODCE - C++ Libraries
CSS	DCCI	Developer Support	COTS/Custom				4	OODCE - Encapsulation
CSS	DCCI	Thread Service	COTS/Custom	500	4	n/a	2	
CSS	DCCI	Directory/ Naming Service	COTS/Custom	1,000	4	5,500		
CSS	DCCI	Object Passing	COTS/Custom	n/a		1,500	4	RPCs
CSS	DCCI	Object Passing	COTS/Custom				2	RPC pipes
CSS	DCCI	Event Service	COTS/Custom	n/a		6,000	4	
CSS	DCCI	Life Cycle Service	COTS/Hybrid	n/a		4,000	4	
CSS	DCCI	Security Service	COTS/Custom	1,500	4	7,500	4	
MSS	MCI	Accountability Mgt	COTS			2,000		User Registration
MSS	MCI	Accountability Mgt	COTS	n/a		1,000		Accountability
MSS	MCI	Security Management	COTS/Hybrid	500	4			DCE Authentication only
MSS	MCI	Security Management	Hybrid			1,500		Compliance Management
MSS	MCI	Security Management	COTS			0	2	Audit Information Collection
MSS	MCI	Security Management	COTS/Custom/Hybrid			1,500		Intrusion Detection
MSS	MCI	Security Management	COTS			0	1	Reporting
MSS	MCI	Maps & Collections	COTS/Hybrid	500		500	4	
MSS	MCI	Management Framework	COTS/Hybrid	500	2	1,000	4	Discovery & Monitoring in Rel . A
MSS	MCI	Fault Management	COTS/Hybrid	500	4		2	Fault Identification, Isolation
MSS	MCI	Fault Management	COTS/Hybrid	n/a		500	2	Fault Recovery
MSS	MCI	Fault Management	COTS/Hybrid	n/a		500	1	Fault Reporting
MSS	MCI	Performance Management	COTS		4			
MSS	MCI	Performance Management	COTS/Custom/Hybrid	n/a		2,000	2	Analysis
MSS	MCI	Performance Management	Custom	n/a		500	2	Monitoring
MSS	MCI	Performance Management	COTS/Hybrid	n/a		500	1	Reporting
MSS	MCI	Management User Interface	COTS		2		4	
MSS	MCI	Scheduling	COTS				4	
MSS	MCI	Trouble Ticketing	COTS				2	
MSS	MCI	Physical CM	COTS				2	
MSS	MCI	Management Data Access	Custom/Hybrid	n/a	4	4,000	4	

Subsystem	CI	Component	Dev. Type	Ir1		Rel. A		Remarks
MSS	MCI	DBMS	COTS/Custom/Hybrid	0	2	2,000	2	Schema/ ad-hoc Reporting
MSS	MCI	Startup & Shutdown	Hybrid	NSP				p/o Mgmt Frwk Development
MSS	MCI	OA Tools	COTS	0	1		4	OA Products
MSS	MCI	OA Tools	COTS				4	Ops Management Tools
MSS	MACI	Management Agents	COTS/Custom /Hybrid	500	4	3,500	4	Agent
MSS	MACI	Management Agents	COTS/Custom /Hybrid		4		4	MIB
MSS	MLCI	Configuration Management	COTS/Hybrid	500		1,500	4	
ISS	INCI	Transport Service - TCP	COTS	n/a	1	n/a		
ISS	INCI	Transport Service - UDP	COTS	n/a	1	n/a		
ISS	INCI	Transport Service - Other	COTS	n/a		n/a		Rel. B
ISS	INCI	Network Services - IP	COTS	n/a	1	n/a		
ISS	INCI	Network Services - ICMP	COTS	n/a	1	n/a		
ISS	INCI	Network Services - ARP	COTS	n/a	1	n/a		
ISS	INCI	Datalink/Physical - FDDI	COTS	n/a		n/a	1	
ISS	INCI	Datalink/Physical - HiPPI	COTS	n/a		n/a		Rel. B
ISS	INCI	Datalink/Physical - ATM	COTS	n/a		n/a	4	
ISS	INCI	Datalink/Physical - Ethernet	COTS	n/a		n/a	1	
ISS	INCI	Datalink/Physical - SMDS	COTS	n/a		n/a	1	
ISS	INCI	Datalink/Physical - PPP	COTS	n/a		n/a	1	
ISS	INCI	Datalink/Physical - SLIP	COTS	n/a		n/a		Rel. B

This page intentionally left blank.

Abbreviations and Acronyms

ACL	Access Control List
ADC	Affiliated Data Center
ATM	Asynchronous Transfer Model
CCB	Change Control Board
CCR	Change Control Request
CDRL	Contract Data Requirements List
CI	Configuration Item
COTS	Commercial off-the-shelf (hardware or software)
CSMS	Communications and System Management Segment
CSS	Communications Subsystem
CUT	Code and Unit Test
DAAC	Distributed Active Archive Center
DBMS	Data Base Management System
DCCI	Distributed Computing Configuration Item
DCE	Distributed Computing Environment
DCHCI	Distributed Communications Hardware Configuration Item
DFS	Distributed File System
DID	Data Item Description
DNS	Domain Name Services
Ecom	EOS Communications
ECS	EOSDIS Core System
EDC	EROS Data Center
EDF	ECS Development Facility
EDOS	EOS Data and Operations System
EOC	Earth Observation Center; EOS Operations Center
EOSDIS	EOS Data and Information System
EP	Evaluation Package
ESN	EOSDIS Science Network

FDDI	Fiber Distributed Data Interface
FOS	Flight Operations Segment
FTP	File Transfer Protocol
GFE	Government Furnished Equipment
GSFC	Goddard Space Flight Center
HIPPI	High Performance Parallel Interface
HLL	High Level Language
IDL	Interactive Data Language
IDR	Intermediate Design Review
ILN	Intermediate Logic Network
INCI	Internetworking Configuration Item
INHCI	Internetworking Hardware Configuration Item
IRD	Inperface Requirements Document
ISS	Internetworking Subsystem
IST	Instrument Support Terminal
IV&V	Independent Verification and Validation
LAN	Local Area Network
Landsat	Land Remote-Sensing Satellite
LaRC	Langley Research Center
MACI	Management Agents Configuration Item
M&O	Maintenance and Operations
MCI	Management Software Configuration Item
MDT	Mean Downtime
MHCI	Management Hardware Configuration Item
MIB	Management Information Base
MLCI	Management Logistics Configuration Item
MSFC	Marshall Space Flight Center
MSS	Management Subsystem
MUI	Management User Interface
NOAA	National Oceanic and Atmospheric Administration
NSI	NASA Science Internet

ODC	Other Data Center
OO	Object Oriented
OSF	Open Systems Foundation
OSI	Open System Interconnect
PDR	Preliminary Design Review
POSIX	Portable Operating System Interface for Computer Environments
PSCN	Program Support Communications
RIP	Routing Information Protocol
RMA	Reliability, Maintainability, Availability
SCF	Science Computing Facility
SDPS	Science Data Processing Segment
SDR	System Design Review
SLOC	Source Lines of Code
SM	Staff Month
SMC	System Management Center
SNMP	Simple Network Management Protocol
SRS	Software Requirements Specification
TRMM	Tropical Rainfall Measuring Mission
TSDIS	TRMM Science Data and Information System
WAN	Wide Area Network